CLAIMS

What is claimed is:

1. 1. A system comprising:

2. a data filter coupled to a text-to-image converter for

3. converting filtered data into a set of digital images, the set of

4. digital images being defined as a set of texture maps; and

5. a memory analyzer for analyzing set-top box layout and

6. indicating available memory types, the memory analyzer being

7. coupled to a memory distributor, the memory distributor for

8. distributing texture maps.

1. 2. The system of claim 1, wherein a total size of the set

2. of texture maps is a sum of all texture map sizes.

1. 3. The system of claim 2, further comprising:

2. a processor coupled to the data filter, the processor

3. executing a first logic in which the total size of the set of

4. texture maps is less than or equal to a memory size; and

5. a second logic if the total size of the set of texture

6. maps is greater than the memory size, then dividing the set of

7. texture maps into at least two groups.

1. 4. The system of claim 3, wherein a total size of the first

2. group is the largest possible sum of texture map sizes for which

3. the total size of the first group is less than the memory size.

5. The system of claim 3, wherein a total size of the second group is the difference between the total size of the set of texture maps and the total size of the first group.

6. The system of claim 3, wherein the set of texture maps of the first group is stored in a first memory.

7. The system of claim 3, wherein the set of texture maps of the second group is stored in a second memory.

8. The system of claim 3, wherein the set of texture maps of the second group are compressed to fit into the first memory.

9. The system of claim 8, further comprising a compression engine.

10. A method comprising:

computing a total size of a set of texture maps;

comparing the total size of the set of texture maps with a memory size;

dividing the set of texture maps into at least two groups if the total size of the set of texture maps is larger than the memory size, such that the total size of the texture maps in a first group is the largest possible sum of texture map sizes for which the total size of texture maps in the first group is less than the memory size.

1   11.   The method of claim 10 wherein computing a total size of

2   a set of texture maps comprises:

3   computing a sum of all texture map sizes.

1   12.   The method of claim 10 further comprising:

2   storing the set of texture maps in a first memory if the

3   total size of the set of texture maps is less than or equal to the

4   first memory size.

1   13.   The method of claim 10 further comprising:

2   storing a first group of texture maps in a first memory.

1   14.   The method of claim 10 further comprising:

2   storing a second group of texture maps in a second

3   memory.

1   15.   The method of claim 14 further comprising:

2   compressing the second group of texture maps to fit into

3   C memory if B memory is not available.